

# Breakpoint Maintenance

A breakpoint causes the execution of a Natural object to be interrupted at a predefined statement line. This section describes how and when to set breakpoints. Note that the maintenance functions described here may also be invoked from an object source by using the function List Object Source.

Below is information on:

- Conditions of Use
  - Activate Breakpoint
  - Deactivate Breakpoint
  - Delete Breakpoint
  - Display Breakpoint
  - Modify Breakpoint
  - Set Breakpoint
- 

## Conditions of Use

A breakpoint is defined by specifying the name of the Natural object to be processed and the line number of the object's source code where the breakpoint is to be executed.

Once a breakpoint has been specified, it remains set for the entire Natural session, unless you delete it.

A breakpoint refers to a specific line number in a source code. A subsequent change of the source code itself may therefore lead to the breakpoint no longer applying to the desired statement, and thus the program not being interrupted at the desired position. To circumvent this problem with program loops, labels can be set within these loops. Breakpoints set for these labels are adjusted to the correct line number if statement lines are inserted or deleted.

The unique identifier for a breakpoint is the Spy Number as assigned by the Natural Debugger.

Breakpoints cannot be set in copycode, on comment lines, on any statement line other than the first one (if a single statement occupies more than one program line), and on lines that contain one of the following statements only:

- AT BREAK OF
- AT END OF DATA
- AT END OF PAGE
- AT START OF DATA
- AT TOP OF PAGE
- BEFORE BREAK
- DECIDE  
(on lines with WHEN and VALUE clauses, however, breakpoints can be set)
- DEFINE SUBROUTINE
- DEFINE WINDOW
- FORMAT
- IF NO RECORDS FOUND
- ON ERROR
- OPTIONS

Whether it is possible or not to set breakpoints for lines compiled with the Natural Optimizer Compiler depends on the NODBG option of the OPTIONS statements as described in "Switching on the Optimizer Compiler" in the Natural Optimizer Compiler documentation.

## Activate Breakpoint

### ▶ To set the current state of specified breakpoints to "active"

- On the the Breakpoint Maintenance menu, enter Function Code **A**, an object name and/or a line number.  
Or enter the direct command `ACTIVATE BREAKPOINT object and/or line`  
(see also the syntax of `ACTIVATE` in Command Summary and Syntax).

If you do not specify an object name or a line number, **all** breakpoints are activated.

## Deactivate Breakpoint

### ▶ To set the current state of specified breakpoints to "inactive"

- On the the Breakpoint Maintenance menu, enter Function Code **B**, an object name and/or a line number.  
Or enter the direct command `DEACTIVATE BREAKPOINT object and/or line`  
(see also the syntax of `DEACTIVATE` in Command Summary and Syntax).

If you do not specify an object name or a line number, **all** breakpoints are deactivated.

## Delete Breakpoint

### ▶ To delete specified breakpoints

- On the the Breakpoint Maintenance menu, enter Function Code **C**, an object name and/or a line number.  
Or enter the direct command `DELETE BREAKPOINT object and/or line`  
(see also the syntax of `DELETE` in Command Summary and Syntax).

If you do not specify an object name or a line number, **all** breakpoints are deleted.

## Display Breakpoint

### ▶ To display specified breakpoints

- On the Breakpoint Maintenance menu, enter Function Code **D**, an object name and/or a line number.  
Or enter the direct command `DISPLAY BREAKPOINT object and/or line`  
(see also the syntax of `DISPLAY` in Command Summary and Syntax).

If a unique breakpoint has been specified, the Display Breakpoint screen appears and all specifications of this breakpoint are displayed. The Display Breakpoint screen is identical to the Modify Breakpoint screen. For an explanation of the fields, see Display/Modify Breakpoint Screen below.

If no unique breakpoint has been specified, a list displays all breakpoints set for the current environment. On the list, you can activate, deactivate, display, modify or delete a breakpoint by marking it with the line command `AC`, `DA`, `DI`, `MO` or `DE` respectively.

If you do not specify an object name or a line number, **all** breakpoints are displayed.

## Modify Breakpoint

### ▶ To modify specified breakpoints

1. On the Breakpoint Maintenance menu, enter Function Code **M**, an object name and/or a line number.  
Or enter the direct command **MODIFY BREAKPOINT** *object* and/or *line*  
(see also the syntax of **MODIFY** in Command Summary and Syntax).  
If a unique breakpoint has been specified, the Modify Breakpoint screen appears and the breakpoint specifications can be modified. The Modify Breakpoint screen is identical to the Display Breakpoint screen. For an explanation of the fields, see Display/Modify Breakpoint Screen below.  
If no unique breakpoint has been specified, a list displays all breakpoints set for the current environment.  
On the list, you can activate, deactivate, display, modify or delete a breakpoint by marking it with the line command **AC**, **DA**, **DI**, **MO** or **DE** respectively.
2. On the Modify Breakpoint screen, choose **PF3/Exit** or **PF5/Save** to save any modification.  
If you choose **PF12/Canc**, the breakpoint remains unchanged.

If you do not specify an object name or a line number, **all** breakpoints are displayed for selection and modification: see Display Breakpoint above.

Below is information on:

- Display/Modify Breakpoint Screen

## Display/Modify Breakpoint Screen

The Modify Breakpoint screen provides the following fields:

Field	Explanation
Spy Number	A unique number assigned by the Natural Debugger when setting the breakpoint.
Initial State	Specifies the current state of the breakpoint: active or inactive.
Breakpoint Name	Valid input: range from 1 to 12 characters. The default name for breakpoints consists of the object name and the line number.
Object Name	The name of an object available in the current library or one of its steplibs. The default name is the name of the default object (see the section Start the Natural Debugger) if defined.
Line Number	The line number of a statement in the object source code. See also Conditions of Use above. You can also specify BEG, END or ALL as line numbers:  <div style="margin-left: 40px;"> BEG      The breakpoint is to be executed at the first statement executed.  END      The breakpoint is to be executed at the last statement executed.  ALL      The breakpoint is to be executed for each program line of the object specified. </div>
Label	Refers to a label set earlier in the source code of an object for statements that define processing loops: see also Conditions of Use above. Valid input: range from 1 to 32 characters.
Skips before Execution	Determines that the breakpoint is not to be executed until the corresponding statement line has been executed a certain number of times. Valid input: range from 0 (default) to 32767.
Number of Executions	Any value greater than zero (0) determines the maximum number of breakpoint executions. Valid input: range from 0 (default) to 32767.
Commands	Up to six debug commands. Enter one command per line. For a summary of all available commands, see Command Summary and Syntax.  <b>Attention:</b> If you delete the command BREAK when modifying a breakpoint and you do not enter any command that issues a dialog, there is no way for the Natural Debugger to receive control during program interruption.

## Set Breakpoint

### To add a breakpoint to a session

- On the Breakpoint Maintenance menu, enter Function Code S, an object name and/or a line number.  
Or enter the direct command SET BREAKPOINT *object* and/or *line*  
(see also the syntax of SET in Command Summary and Syntax).

If object name and line number are specified correctly, the breakpoint is set immediately and a corresponding confirmation message is displayed on the screen. The breakpoint receives the default command (BREAK), its initial and current state are set to "active" and no execution restrictions are specified. Note that if you delete the command

**BREAK** when setting a breakpoint and you do not enter any command that issues a dialog, there is no way for the Natural Debugger to receive control during program interruption.

If you specify not an object name but a valid line number, the name of the default object (see the section **Start the Natural Debugger**) is assumed and the breakpoint is also set immediately. If there is no default object defined, a selection window appears that displays all objects available in the current library.